

Magician's Corner: 7. Using Convolutional Neural Networks to Reduce Noise in Medical Images

Nathan Robert Huber, BSc • Andrew D. Missert, PhD • Bradley J. Erickson, MD, PhD

From the Department of Radiology, Mayo Clinic, 200 First St SW, Rochester, MN 55905. Received March 16, 2020; revision requested April 21; revision received June 11; accepted June 16. Address correspondence to B.E. (e-mail: bje@mayo.edu).

Conflicts of interest are listed at the end of this article.

Radiology: Artificial Intelligence 2020; 2(5):e200036 • <https://doi.org/10.1148/ryai.2020200036> • Content code: AI • ©RSNA, 2020

Medical imaging is driven to produce the best possible images while reducing the radiation dose or acquisition time. Normally, this trade-off is dealt with by using the best possible detection systems and experimenting with different acquisition techniques. Recently, deep learning methods have been applied to images acquired with low dose (or less acquisition time in the case of MRI) to produce images that appear similar to full-dose images. Various deep learning techniques to improve image quality and reduce acquisition time or dose have been reviewed (1). In this article, we describe the development of such a machine learning system, with the specific application to low-dose CT, although the principles apply to other radiologic images, including projection radiography, MRI, and PET.

As with other articles in this series, please begin by loading the notebook using Colab (<https://colab.research.google.com>, File > Open > Github, and search for RSNA, and then find “CNNDenoisingTutorial_Magicians-Corner” or by clicking on this link: https://github.com/RSNA/MagiciansCorner/blob/master/CNNDenoisingTutorial_MagiciansCorner.ipynb). Once open, please execute cell 1, which loads in the TensorFlow routines and a few others that we need.

Reducing noise in an image can be approached as a learning problem, assuming that one has many pairs of low-noise (ie, full-dose) images with corresponding noisy (ie, low-dose) images. Our goal is to then train a network to make a noisy image into a low-noise image.

It is worth noting that this problem is ill-posed. We should not expect to reconstruct the signal exactly, but we can make a prediction that agrees with the observed noisy data and prior knowledge. During optimization, we aim to have the convolutional neural network (CNN) encode meaningful prior knowledge from many training examples into the predictions.

In the example here, there is perfect correspondence between full-dose and low-dose images (other than noise) because the CT images were collected once at full dose, and noise was added to simulate low-dose acquisition (2). Figure 1 (also the content just after cell 1) visually describes this process, and the source of the data used. Cell 2 loads this data: please execute that cell, and then also run cell 3 which organizes the data.

Cell 4 defines a function for showing CT images, with the default window setting being for soft tissue. Note that you can adjust the specific window-level values in this code, and when you call this function, you can override

‘soft tissue’ setting with ‘bone’ or ‘lung’ if you wish. Run cell 4. Cell 5 uses that function to display some of the images, including the low-dose, routine-dose, and the difference between them. The image reconstruction process introduces a characteristically streaky and locally correlated noise texture, which can be seen in the difference images. Please run cell 5.

The next step is to define a simple model that we hope will learn how to filter images (hereafter referred to as “reduced noise”) that look like routine-dose (“low-noise”) from low-dose (“noisy”) images. As shown in the information cell below cell 5 (see Fig 2), we will use five convolutional layers. Note that unlike prior articles, we do not use pooling layers because we don’t want to reduce resolution. Because the model only uses convolutional layers, it can operate on images with arbitrary pixel size. In this notebook, we will train the model using image patches with a size of 64×64 pixels, but test the model using full-size 512×512 CT images. Cell 6 defines the various hyperparameters such as the number of layers, the number of filters, the kernel size, the stride (how many pixels the convolution kernel moves each time), and the activation function (‘relu’ = rectified linear unit). Run cell 6 to build the model.

Cell 7 sets a few more hyperparameters related to training the model, including the learning rate, optimizer, batch size, and number of epochs. It also creates a place to store images as the model is trained so we can review its progress. Finally, it trains the model, and of note, the loss function is the mean squared error (MSE), which is the square of the difference between pixel values in the low-noise image and the reduced-noise image. It makes sense to use such a pixel versus pixel error function since we don’t have a class to predict or a segmentation contour to compare against. MSE is a popular error metric, as it penalizes large deviations from the desired values more than the mean error function. Run cell 7 to train your model, and then run cell 8 to see the results at different steps in the training process. Note how the quality of the output images changes as the model is trained. Cell 9 displays images that we set aside in our test set, so run cell 9 to display the test images. You should see images that exhibit lower noise, based both on the difference map (you will see some structure in the difference map indicating a bit of lost signal) and when looking at the magnified image.

The choice of loss function has a substantial impact on the denoising result. MSE is a popular metric, but it

Summary

This article shows how to train a convolutional neural network to reduce noise in CT images, although the principles apply to medical and nonmedical images; authors also explore mathematical and visually weighted loss functions to adjust the appearance.

Key Points

- In this article, authors show how to train a convolutional neural network to reduce noise on medical images, especially low-dose CT images from the recent American Association of Physicists in Medicine low-dose challenge dataset.
- Human visual feature weighting can be used as a part of the loss term to improve the visual appearance of the filtered images.

focuses only on the average difference between corresponding pixels, and that doesn't necessarily reflect the things in images that we humans notice, or the things that are most important for the clinical task at hand. Choosing the optimum loss function is an area of ongoing research (3,4). For this demonstration, we add a component to the error metric that includes image features that are important parts of what humans perceive in images. Ideally, this "feature loss" will better quantify the perceptual differences between two images, thus making the reduced-noise images appear better. Instead of defining these features ourselves, we will use part of a pretrained CNN for image classification, in this case the popular VGG16. Run cell 10 to load in the VGG16 model, and then run cell 11 to extract

the perceptual layers from the full VGG16 model. Note you can experiment with features extracted from different layers in the model by adding them to the list at the top of this cell. Figure 3 shows a schematic of the feature loss approach, where the first three blocks of VGG16 were applied to the CNN output and target image to extract features from the images. The feature contents were then compared using MSE. One advantage of using feature loss is improved retention of realistic CT texture within the CNN output.

In cell 13, the visual importance of various features from VGG16 are shown, using a CT image as an example. Run cell 13 to see areas of lighter color reflect areas of the CT image that match the features that VGG16 found to be important.

In cell 14, we define a new loss function that adds the VGG16 features to the MSE loss. Note that a scaling factor has been introduced that reflects both the typical magnitudes of each term and their relative importance. This new error function is then used to train the *model_vggloss* defined in cell 15. Run cells 14 and 15 if you haven't already. Then run cell 16 to train this new model. Once that is complete, run cell 17 to display the input low-dose image, the MSE-optimized model output, and the output of the combined MSE and feature loss model (shown in Fig 4). Some people may prefer the result with the added feature loss, noting that some details may be more apparent despite the increased noise. Depending on the error function used, the MSE image is "closer" to the full-dose image (if one uses the MSE to

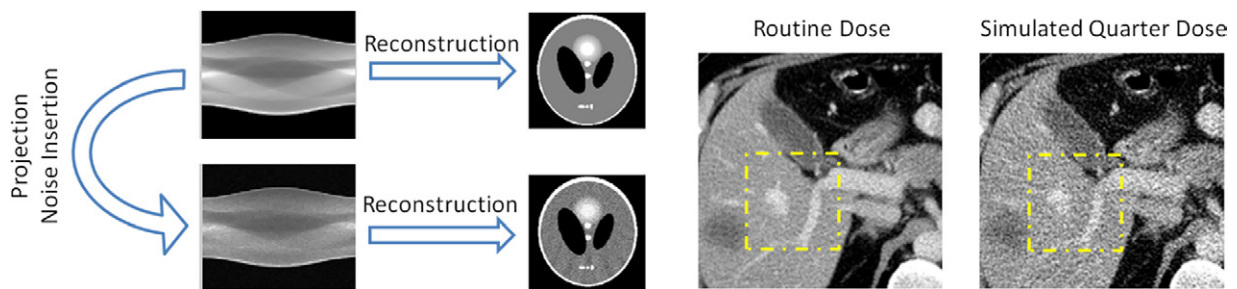


Figure 1: Schematic diagram of data used. Standard sinogram data collected at full dose are reconstructed in a normal fashion to produce the reference full-dose image. The sinogram data also have a copy made that has noise injected into it, modeling what a lower-dose sinogram image would look like. This is then reconstructed, producing a simulated low-dose image, but there is exact correspondence of the structures in the images, which is important for training purpose.

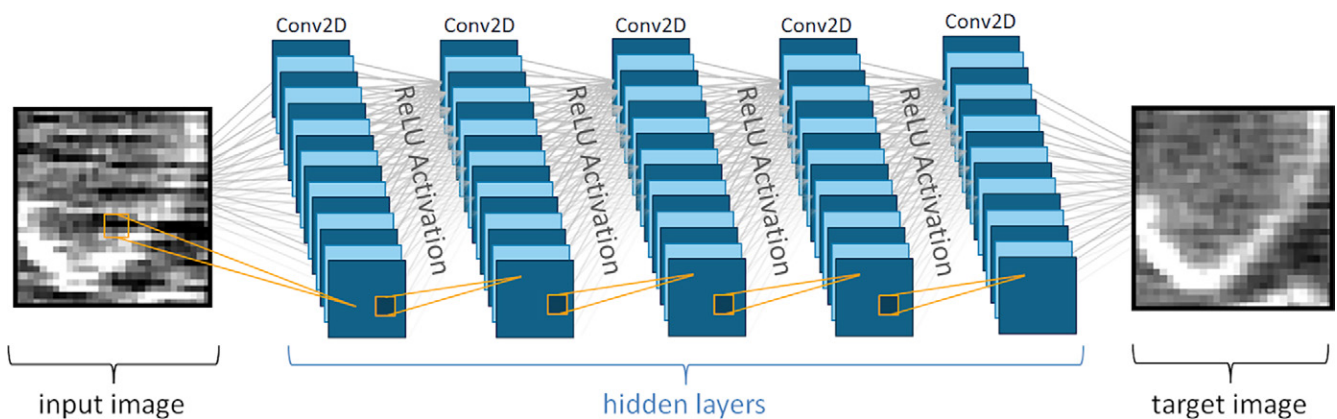


Figure 2: Schematic of the simple model architecture used for denoising. This model employs five layers of two-dimensional convolutions. Each layer will have an associated activation function (rectified linear unit [ReLU] in this case) that adds nonlinearity to the filtering function. There is no pooling, and thus the output resolution matches the input resolution.

measure similarity), but in this case, the mathematically more similar image may not be the visually preferred image.

We trained two very simple models to filter images using a somewhat limited set of images in just a short time. One can

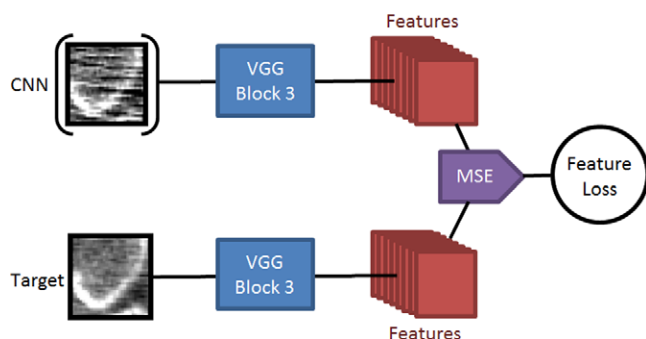


Figure 3: Schematic of the feature loss term. The first three blocks of VGG16 were applied to the convolutional neural network (CNN) output and target image to extract features from the images. The feature contents were then compared using mean squared error (MSE). One advantage of using feature loss is improved retention of realistic CT texture within the CNN output.

further improve the results with a more complex model and an improved training and optimization framework (5). The last cell (cell 18) compares the output of such an optimized method (described in Proceedings of RSNA 2019, “Patient-Specific Noise Reduction Using a Deep Convolutional Neural Network” [SSE24–02], <http://archive.rsna.org/2019/PhysicsandBasicScience.pdf>) versus the model we trained here. These more advanced methods allow greater preservation of anatomic details after noise reduction.

Summary

In this article, we applied CNNs to the task of reducing noise in simulated low-dose CT images. Compared with traditional image processing methods, deep learning uses examples of the noisy images plus the true low-noise images, and the CNN learns how to make the low-dose (noisy) look as good as the full-dose (“low noise”) images. No noise-reduction algorithm is perfect, and at some noise level, image features will be irreversibly lost. Because of this, subtle abnormalities may be removed

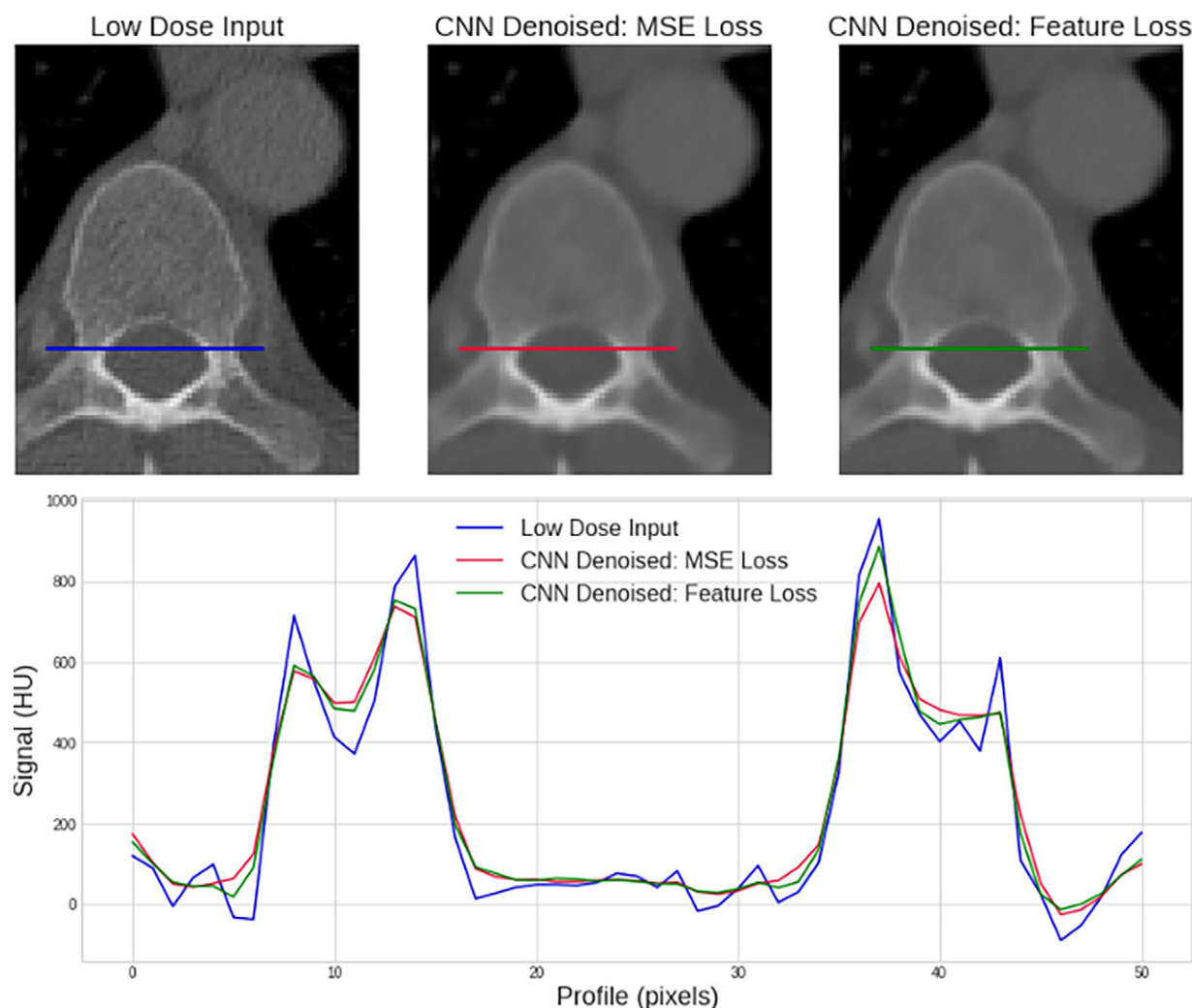


Figure 4: A comparison of the original low-dose image (left), the image denoised using the simple convolutional neural network (CNN) with mean squared error (MSE) as the loss function (middle), and image denoised using the same model with a combination of MSE and VGG16 features as the loss function (right). Although the model architecture and the training data are the same, the different loss function has a significant impact on the quality of the output image. A line profile was included to assess the network’s ability to retain bone structure after noise reduction.

from noise-reduced low-dose CT images relative to routine-dose CT. However, the ability of CNNs to leverage prior information from many training examples makes this approach very powerful, with performance comparing favorably to other established methods (6). Furthermore, this general method can be applied to many other types of medical and nonmedical images. Because the error is well-characterized and controllable, and because traditional image processing filters have been applied to the task of filtering noise out of images, CNNs are already being applied in commercial imaging devices, and likely will continue to make important improvements in image quality. Every radiologist and medical physicist should become familiar with this important technology.

Acknowledgments: We acknowledge Cynthia McCollough, PhD, the Mayo Clinic, the American Association of Physicists in Medicine, and grants EB017095 and EB017185 from the National Institute of Biomedical Imaging and Bioengineering for distributing the data used within this publication.

Author contributions: Guarantors of integrity of entire study, N.R.H., A.D.M., B.J.E.; study concepts/study design or data acquisition or data analysis/interpretation, all authors; manuscript drafting or manuscript revision for important intel-

tual content, all authors; approval of final version of submitted manuscript, all authors; agrees to ensure any questions related to the work are appropriately resolved, all authors; literature research, N.R.H., A.D.M.; experimental studies, N.R.H., A.D.M.; and manuscript editing, N.R.H., A.D.M., B.J.E.

Disclosures of Conflicts of Interest: N.R.H. disclosed no relevant relationships. A.D.M. disclosed no relevant relationships. B.J.E. disclosed no relevant relationships.

References

1. Tian C, Fei L, Zheng W, Xu Y, Zuo W, Lin C-W. Deep learning on image denoising: An overview. arXiv [eess.IV]. [preprint] <http://arxiv.org/abs/1912.13171>. Posted 2019. Accessed February 12, 2020.
2. Low Dose CT Grand Challenge. <https://www.aapm.org/GrandChallenge/LowDoseCT/>. Accessed March 10, 2020.
3. Yang Q, Yan P, Zhang Y, et al. Low-Dose CT Image Denoising Using a Generative Adversarial Network With Wasserstein Distance and Perceptual Loss. *IEEE Trans Med Imaging* 2018;37(6):1348–1357.
4. Kim B, Han M, Shim H, Baek J. A performance comparison of convolutional neural network-based image denoising methods: The effect of loss functions on low-dose CT images. *Med Phys* 2019;46(9):3906–3923.
5. Missert AD, Yu L, Leng S, Fletcher JG, McCollough CH. Synthesizing images from multiple kernels using a deep convolutional neural network. *Med Phys* 2020;47(2):422–430.
6. Shan H, Padole A, Homayounieh F, et al. Competitive performance of a modularized deep neural network compared to commercial algorithms for low-dose CT image reconstruction. *Nat Mach Intell* 2019;1(6):269–276.